How do I add custom controls to the common file open or file save dialogs?

devblogs.microsoft.com/oldnewthing/20211227-00

December 27, 2021

Raymond Chen

In the beginning, every program had to create its own File Open and File Save dialogs. The system tried to help a little bit, with list boxes having a `LB_DIR` message to populate the list box from a directory.

Windows 3.1 introduced the common file dialogs, known today as "old-style dialogs", though they were brand new back then. Now developers could use these prewritten dialogs instead of having to write their own. And users got some consistency. Win-win.

The Windows 3.1 common file dialogs could be customized by providing a replacement dialog template. The standard template is provided in the SDK under the name `FileOpen.Dlg`, and you could modify the template, typically by adding new check boxes or other controls, and add the modified template to your component as a custom dialog. Just keep the IDs the same for the pre-existing controls, and don't give your new controls conflicting IDs.

Windows 95 introduced a new style of common dialog. This one uses an Explorer-style interface to select the files. Customization of this type of dialog is done by providing a dialog template not of the entire common file dialog but of just the extra controls you want to add. This style of customization was continued through Windows XP.

Windows Vista redesigned the common file dialogs again, which means another round of "Allow customization of the new thing, but keep doing the old thing when given old customizations." But this time, they were ready. Instead of creating another extension pattern that is tied to the design of the dialog box, the extension pattern is now declarative: You use the `IFileDialogCustomize` interface to add controls to the common file dialogs. The interface also lets you query the state of those controls later. The precise layout of those controls, and the form which they take, is left to the operating system. This makes things (hopefully) more future-proof. Even if the design of the common file dialogs changes again in the future, existing apps which use the declarative customization can still be accommodated.

An example of customizing the file open and save dialogs is provided on MSDN, and there's also a sample on GitHub.

Raymond Chen

Follow